

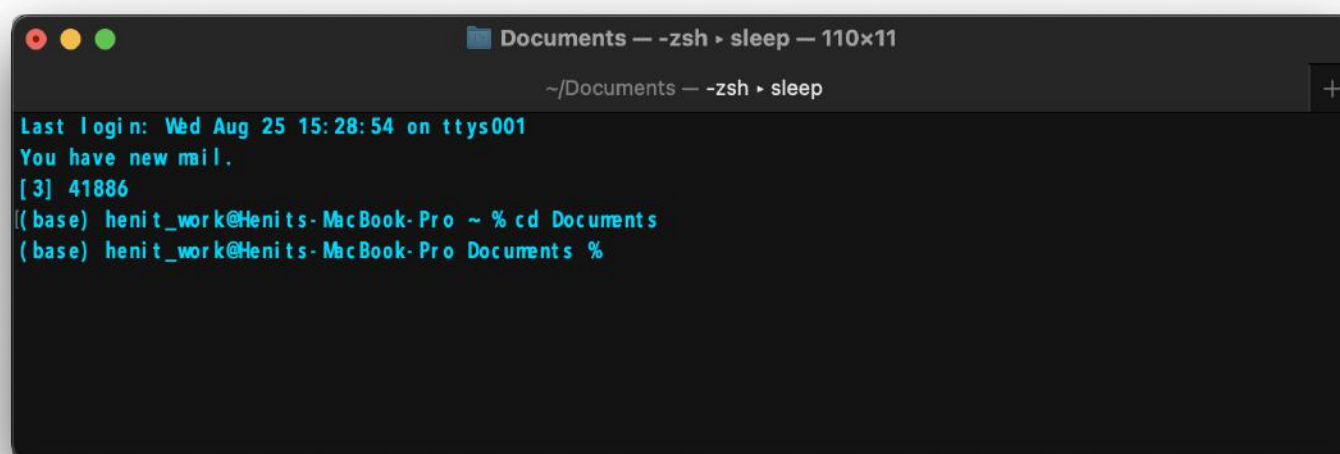
# L-Digital Assignment - 1

Henit Chobisa  
20BDS0316  
C1 - TC1 Slot  
Jagalingam P Sir

## Task 1 - Shell Commands

### **cd :**

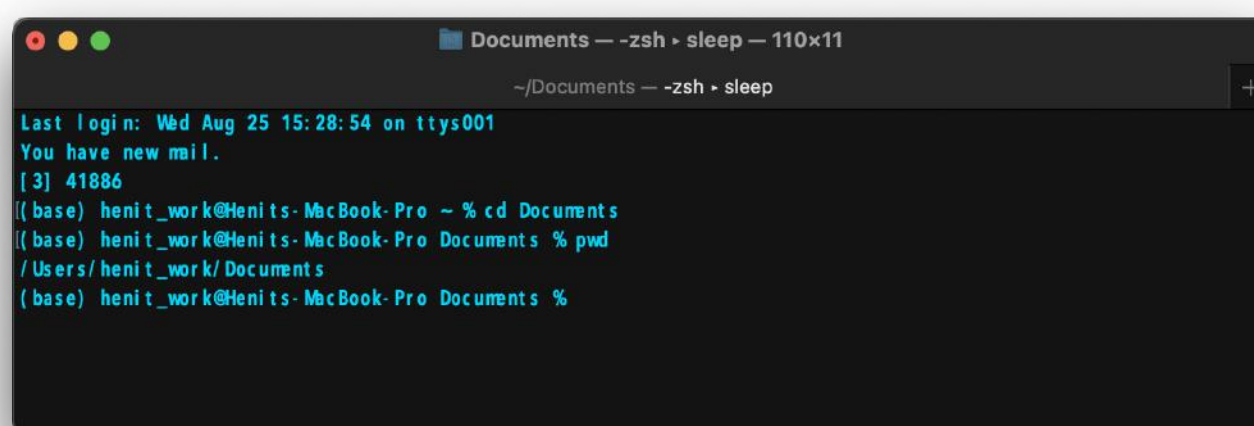
This commands helps us to change the current directory



```
Documents — -zsh • sleep — 110x11
~/Documents — -zsh • sleep
Last login: Wed Aug 25 15:28:54 on ttys001
You have new mail.
[3] 41886
((base) henit_work@Henits-MacBook-Pro ~ % cd Documents
(base) henit_work@Henits-MacBook-Pro Documents %
```

### **pwd :**

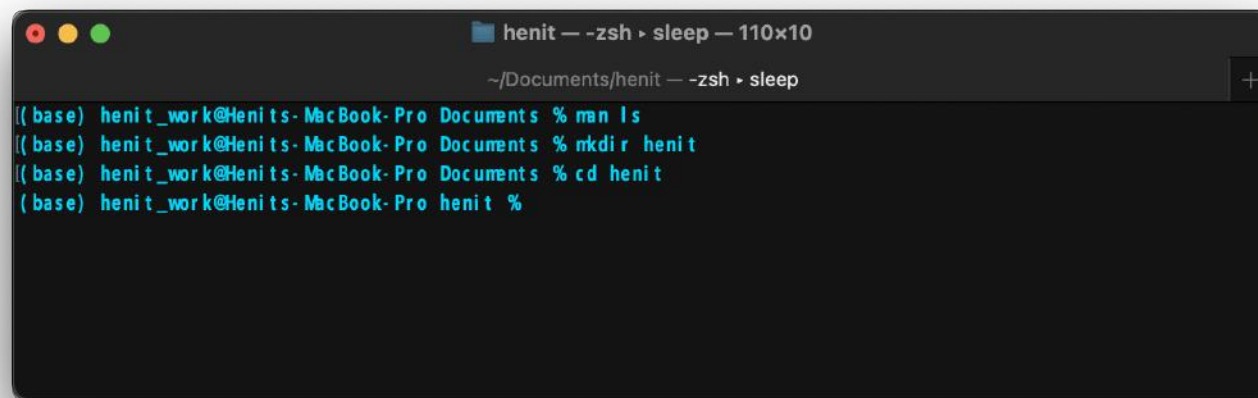
this command helps us in determining the previous the current working directory of the shell.



```
Documents — -zsh • sleep — 110x11
~/Documents — -zsh • sleep
Last login: Wed Aug 25 15:28:54 on ttys001
You have new mail.
[3] 41886
((base) henit_work@Henits-MacBook-Pro ~ % cd Documents
(base) henit_work@Henits-MacBook-Pro Documents % pwd
/Users/henit_work/Documents
(base) henit_work@Henits-MacBook-Pro Documents %
```

### **mkdir :**

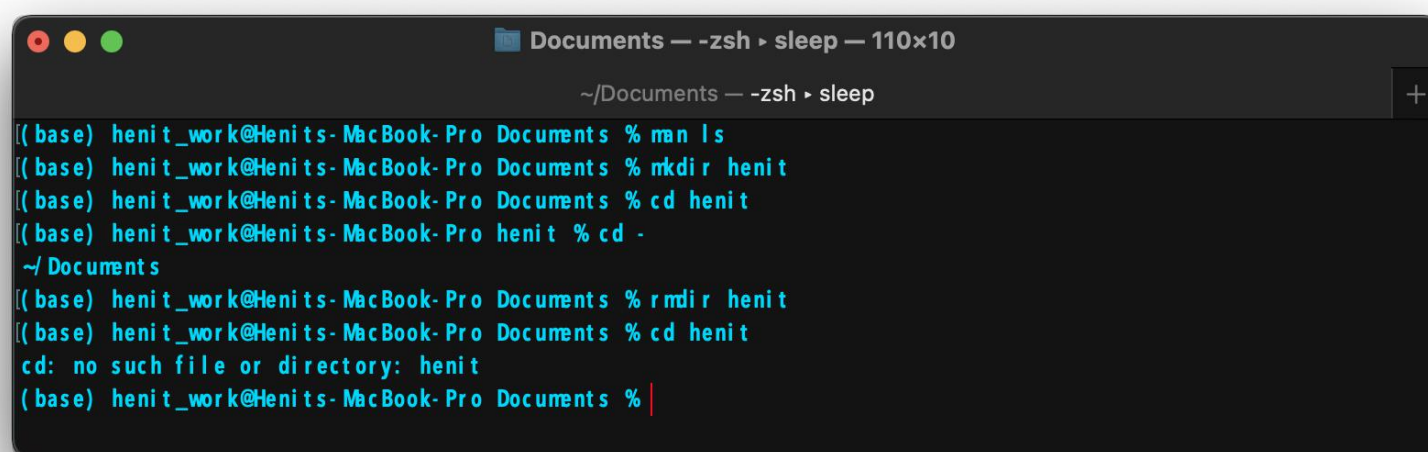
This commands, represents make directory, this can also be used with cd as a followup, so that if the directory is not present, it will create the directory and change the current directory to that particular.



```
henit — -zsh • sleep — 110x10
~/Documents/henit — -zsh • sleep
(( base) henit_work@Henits-MacBook-Pro Documents % man ls
(( base) henit_work@Henits-MacBook-Pro Documents % mkdir henit
(( base) henit_work@Henits-MacBook-Pro Documents % cd henit
(( base) henit_work@Henits-MacBook-Pro henit %
```

## rmkdir :

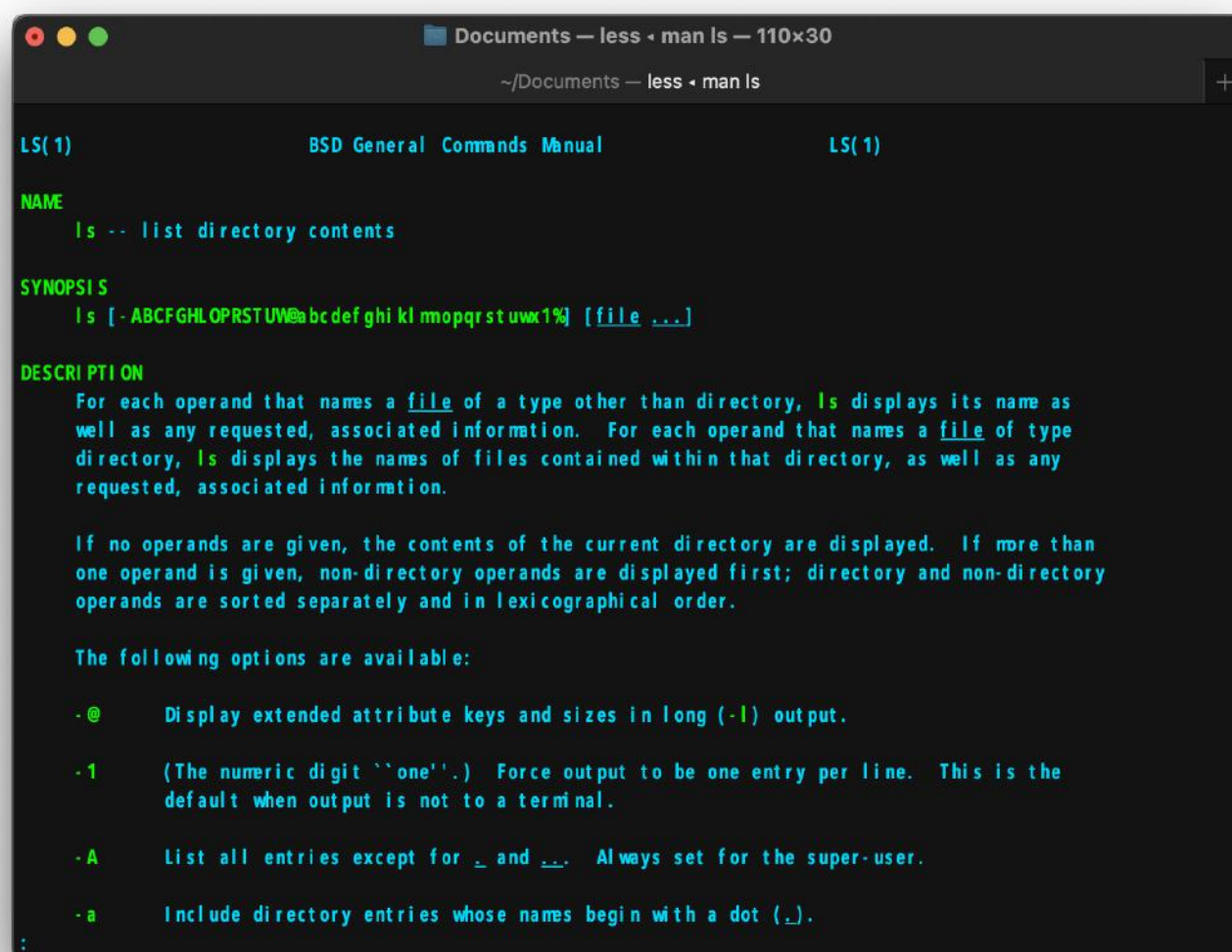
This commands helps in deleting a particular directory from the current directory.



```
Documents — -zsh • sleep — 110x10
~/Documents — -zsh • sleep
(( base) henit_work@Henits-MacBook-Pro Documents % man ls
(( base) henit_work@Henits-MacBook-Pro Documents % mkdir henit
(( base) henit_work@Henits-MacBook-Pro Documents % cd henit
(( base) henit_work@Henits-MacBook-Pro henit % cd -
~/Documents
(( base) henit_work@Henits-MacBook-Pro Documents % rmdir henit
(( base) henit_work@Henits-MacBook-Pro Documents % cd henit
cd: no such file or directory: henit
(( base) henit_work@Henits-MacBook-Pro Documents %
```

## man :

man represents manual, it helps us to get a brief information about any specific command.



```
Documents — less • man ls — 110x30
~/Documents — less • man ls
LS(1)                                BSD General Commands Manual                                LS(1)

NAME
  ls -- list directory contents

SYNOPSIS
  ls [-ABCFGHLOPRSTUW@abcdeghi klmnopqrstuwx1] [file ...]

DESCRIPTION
  For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

  If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

  The following options are available:

  -@      Display extended attribute keys and sizes in long (-l) output.

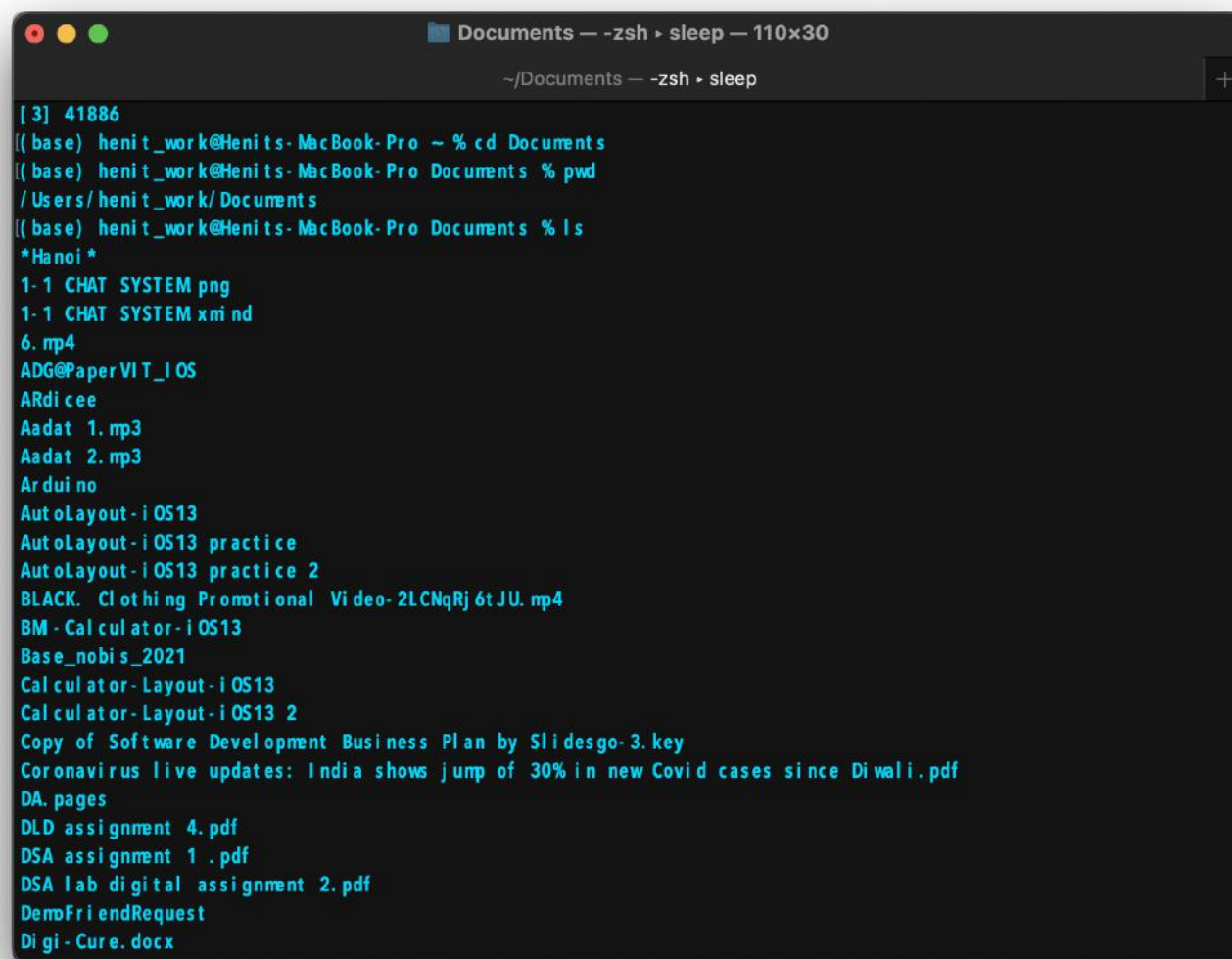
  -1      (The numeric digit ``one''.) Force output to be one entry per line. This is the default when output is not to a terminal.

  -A      List all entries except for . and ... Always set for the super-user.

  -a      Include directory entries whose names begin with a dot (.).
```

## ls :

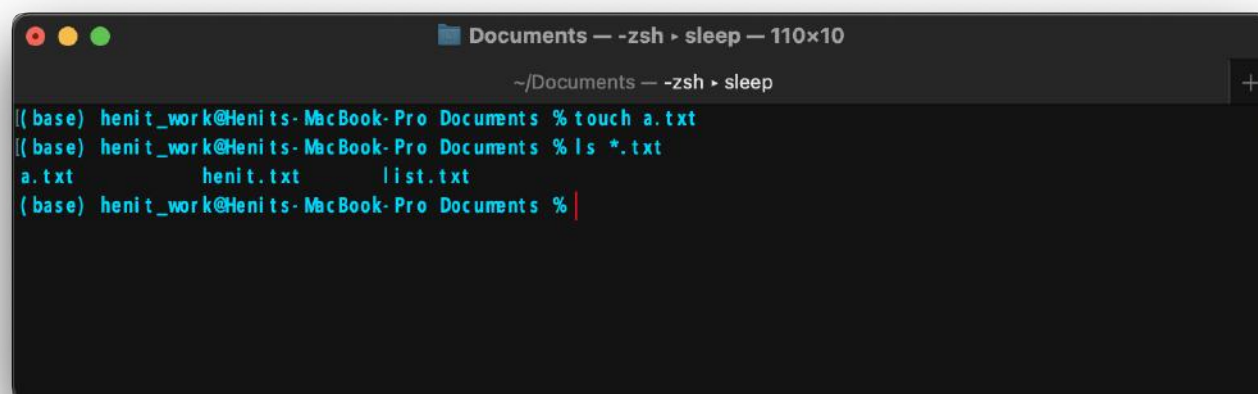
ls or list command lists out all the files in the current directory.



```
Documents — -zsh • sleep — 110x30
~/Documents — -zsh • sleep
[3] 41886
(( base) henit_work@Henits-MacBook-Pro ~ % cd Documents
(( base) henit_work@Henits-MacBook-Pro Documents % pwd
/Users/henit_work/Documents
(( base) henit_work@Henits-MacBook-Pro Documents % ls
*Hanoi*
1-1 CHAT SYSTEM.png
1-1 CHAT SYSTEM.xml
6.mp4
ADG@PaperVIT_IOS
ARdicee
Aadat 1.mp3
Aadat 2.mp3
Arduino
AutoLayout-iOS13
AutoLayout-iOS13 practice
AutoLayout-iOS13 practice 2
BLACK Clothing Promotional Video-2LCNqRj6tJU.mp4
BM-Calculator-iOS13
Base_nobis_2021
Calculator-Layout-iOS13
Calculator-Layout-iOS13 2
Copy of Software Development Business Plan by Slidesgo-3.key
Coronavirus live updates: India shows jump of 30% in new Covid cases since Diwali.pdf
DA.pages
DLD assignment 4.pdf
DSA assignment 1.pdf
DSA lab digital assignment 2.pdf
DemoFriendRequest
Digi-Cure.docx
```

## touch :

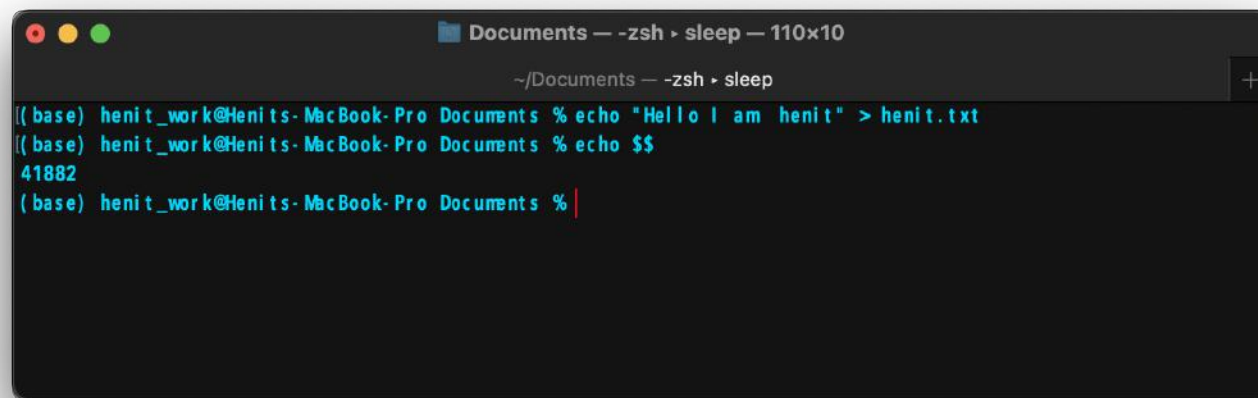
Touch helps us to create a file from the terminal itself, touch file name



```
Documents — -zsh • sleep — 110x10
~/Documents — -zsh • sleep
(( base) henit_work@Henits-MacBook-Pro Documents % touch a.txt
(( base) henit_work@Henits-MacBook-Pro Documents % ls *.txt
a.txt      henit.txt  list.txt
(( base) henit_work@Henits-MacBook-Pro Documents %
```

## echo :

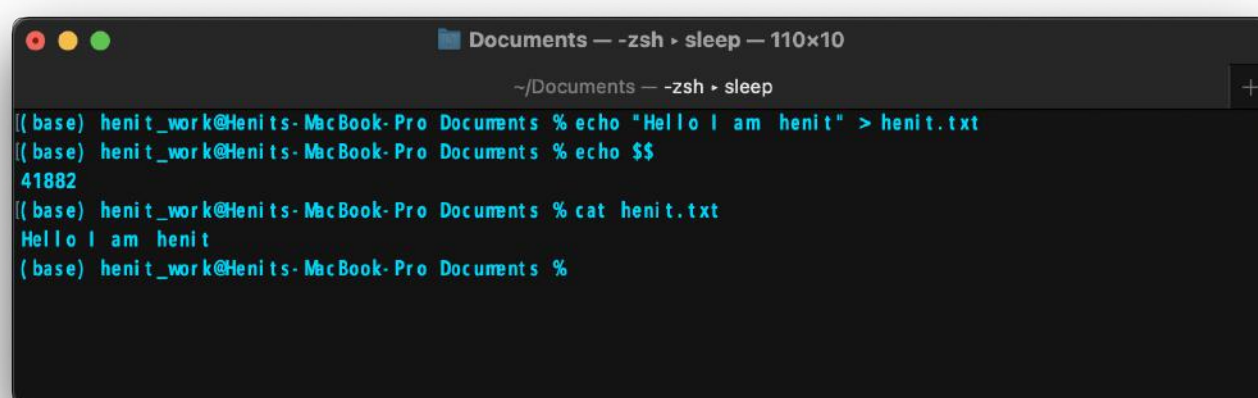
echo works as a print statement in shell, echo \$\$ gets you the process id of the shell

A terminal window titled 'Documents — -zsh > sleep — 110x10' with a subtitle '~ / Documents — -zsh > sleep'. The terminal shows the following commands and output:

```
(( base) henit_work@Henits-MacBook-Pro Documents % echo "Hello I am henit" > henit.txt
(( base) henit_work@Henits-MacBook-Pro Documents % echo $$
41882
(( base) henit_work@Henits-MacBook-Pro Documents %
```

### cat :

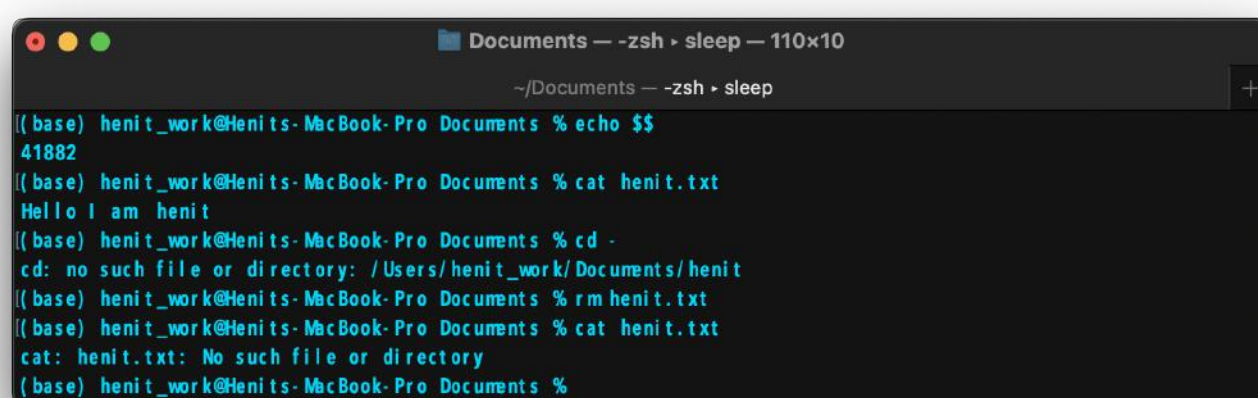
cat helps us to view the files in terminal itself.

A terminal window titled 'Documents — -zsh > sleep — 110x10' with a subtitle '~ / Documents — -zsh > sleep'. The terminal shows the following commands and output:

```
(( base) henit_work@Henits-MacBook-Pro Documents % echo "Hello I am henit" > henit.txt
(( base) henit_work@Henits-MacBook-Pro Documents % echo $$
41882
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit.txt
Hello I am henit
(( base) henit_work@Henits-MacBook-Pro Documents %
```

### rm :

This command followed by the name of the file, helps us to delete a particular file.

A terminal window titled 'Documents — -zsh > sleep — 110x10' with a subtitle '~ / Documents — -zsh > sleep'. The terminal shows the following commands and output:

```
(( base) henit_work@Henits-MacBook-Pro Documents % echo $$
41882
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit.txt
Hello I am henit
(( base) henit_work@Henits-MacBook-Pro Documents % cd -
cd: no such file or directory: /Users/henit_work/Documents/henit
(( base) henit_work@Henits-MacBook-Pro Documents % rm henit.txt
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit.txt
cat: henit.txt: No such file or directory
(( base) henit_work@Henits-MacBook-Pro Documents %
```

### whoami :

whoami, helps us to know the administrator of the computer.



```
Documents — -zsh › sleep — 110x10
~/Documents — -zsh › sleep
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit.txt
Hello I am henit
(( base) henit_work@Henits-MacBook-Pro Documents % cd -
cd: no such file or directory: /Users/henit_work/Documents/henit
(( base) henit_work@Henits-MacBook-Pro Documents % rm henit.txt
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit.txt
cat: henit.txt: No such file or directory
(( base) henit_work@Henits-MacBook-Pro Documents % who am i
henit_work ttys001 Aug 25 16:11
(( base) henit_work@Henits-MacBook-Pro Documents %
```

## cp :

cp command help us to copy a file, cp <file 1> <file 2>

```
Documents — -zsh › sleep — 110x41
~/Documents — -zsh › sleep
(( base) henit_work@Henits-MacBook-Pro Documents % ls *.txt
a.txt          henit2.txt     list.txt
(( base) henit_work@Henits-MacBook-Pro Documents % cat list.txt
Childish
Calm
Flux
Faith
Optimistic
Hopeful
Chill
Blood
Rose
Maple
Hues
Time
Restrain
Instinctive
Incredible
(( base) henit_work@Henits-MacBook-Pro Documents % cp list.txt henit2.txt
(( base) henit_work@Henits-MacBook-Pro Documents % cat henit2.txt
Childish
Calm
Flux
Faith
Optimistic
Hopeful
Chill
Blood
Rose
Maple
Hues
Time
Restrain
Instinctive
Incredible
(( base) henit_work@Henits-MacBook-Pro Documents %
```

## mv :

mv is a representation of move, it helps us to move a file from one directory to another.

```
b — -zsh ▸ sleep — 110x18
~/Documents/test5/b — -zsh ▸ sleep
[ (base) henit_work@Henits-MacBook-Pro test5 % ls
a      b
[ (base) henit_work@Henits-MacBook-Pro test5 % touch henit.txt
[ (base) henit_work@Henits-MacBook-Pro test5 % mv henit.txt b
[ (base) henit_work@Henits-MacBook-Pro test5 % cd b
[ (base) henit_work@Henits-MacBook-Pro b % ls
henit.txt
[ (base) henit_work@Henits-MacBook-Pro b % ]
```

## top :

top command list out all the current processes running in the system

```
b — top — 110x23
~/Documents/test5/b — top
Processes: 547 total, 3 running, 544 sleeping, 2671 threads
Load Avg: 3.72, 3.29, 2.73  CPU usage: 17.1% user, 6.45% sys, 76.52% idle
SharedLibs: 289M resident, 40M data, 35M linkedit.
MemRegions: 166138 total, 1833M resident, 100M private, 842M shared.
PhysMem 8095M used (2277M wired), 94M unused.
VM 3973G vsize, 2305M framework vsize, 20197451(128) swapins, 21496508(0) swapouts.
Networks: packets: 5356456/5074M in, 6489353/1688M out. Disks: 6217333/183G read, 2103989/105G written.

PID  COMMAND      %CPU  TIME    #TH   #WO  #PORT  MEM   PURG   CMPRS  PGRP  PPID  STATE   BOOSTS
42355  com.apple.We 98.1   06:41.14 5/1    1     108   22M    0B     20M   42355 1    running 0[849]
131    WindowServer 9.6    01:49:39 15     5     2456+ 887M+  14M    199M   131    1    sleeping *0[1]
0      kernel_task  8.2    54:36.76 203/8  0      0     518M+  0B     0B     0      0    running 0[0]
51949  top          5.4    00:00.89 1/1    0     28+   4820K+ 0B     0B     51949 41882 running *0[1]
307    TouchBarServ 4.2    08:24.09 6       2     414+  28M+   2688K  12M    307    1    sleeping *0[1]
33596  Microsoft Te 2.7    19:15.21 21     1     271   341M   0B     92M    33588 33588 sleeping *0[6]
396    ControlStrip 2.2    00:16.80 12     7     399+  27M+   0B-    17M    396    1    sleeping *0[1011+]
725    Terminal     2.2    01:04.87 15     6     401+  65M+   32K    18M+   725    1    sleeping *0[2199]
132    tccd         1.8    00:11.31 3       2     54+   3120K+ 64K    1020K- 132    1    sleeping *0[1177+]
185    coreaudiod  1.6    20:14.93 38     2     1619+ 33M+   0B     21M    185    1    sleeping *0[1]
39529  Notion Helpe 1.5    02:59.57 16     1     169   87M+   0B     38M    23211 23211 sleeping *0[1]
211    nsurlsession 1.4    05:16.47 6       5     92    3540K+ 0B     1372K+ 211    1    sleeping 0[957]
134    loginwindow  1.1    00:24.25 3       2     383+  40M    0B     24M    134    1    sleeping *21[850]
177    trustd       1.1    01:07.88 2       1     99+   4788K+ 192K    2148K  177    1    sleeping *0[28743+]
```

## curl :

curl helps us to download a particular file from the internet to our pwd.



```
test5 — -zsh ▸ sleep — 110x23
~/Documents/test5 — -zsh ▸ sleep

(base) henit_work@Henits-MacBook-Pro test5 % ls
a      b
(base) henit_work@Henits-MacBook-Pro test5 % touch output.png
(base) henit_work@Henits-MacBook-Pro test5 % curl https://i.pcmag.com/imagery/reviews/03ai zyl UVApdyLAI ku1AvRV-
39.1605559903.fit_scale.size_760x427.png --output output.png
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left   Speed
100 13387    0 13387    0    0  32257      0 --:--:-- --:--:-- --:--:--  32257
(base) henit_work@Henits-MacBook-Pro test5 % open output.png
(base) henit_work@Henits-MacBook-Pro test5 % ls
a      b      output.png
(base) henit_work@Henits-MacBook-Pro test5 %
```

## stat :

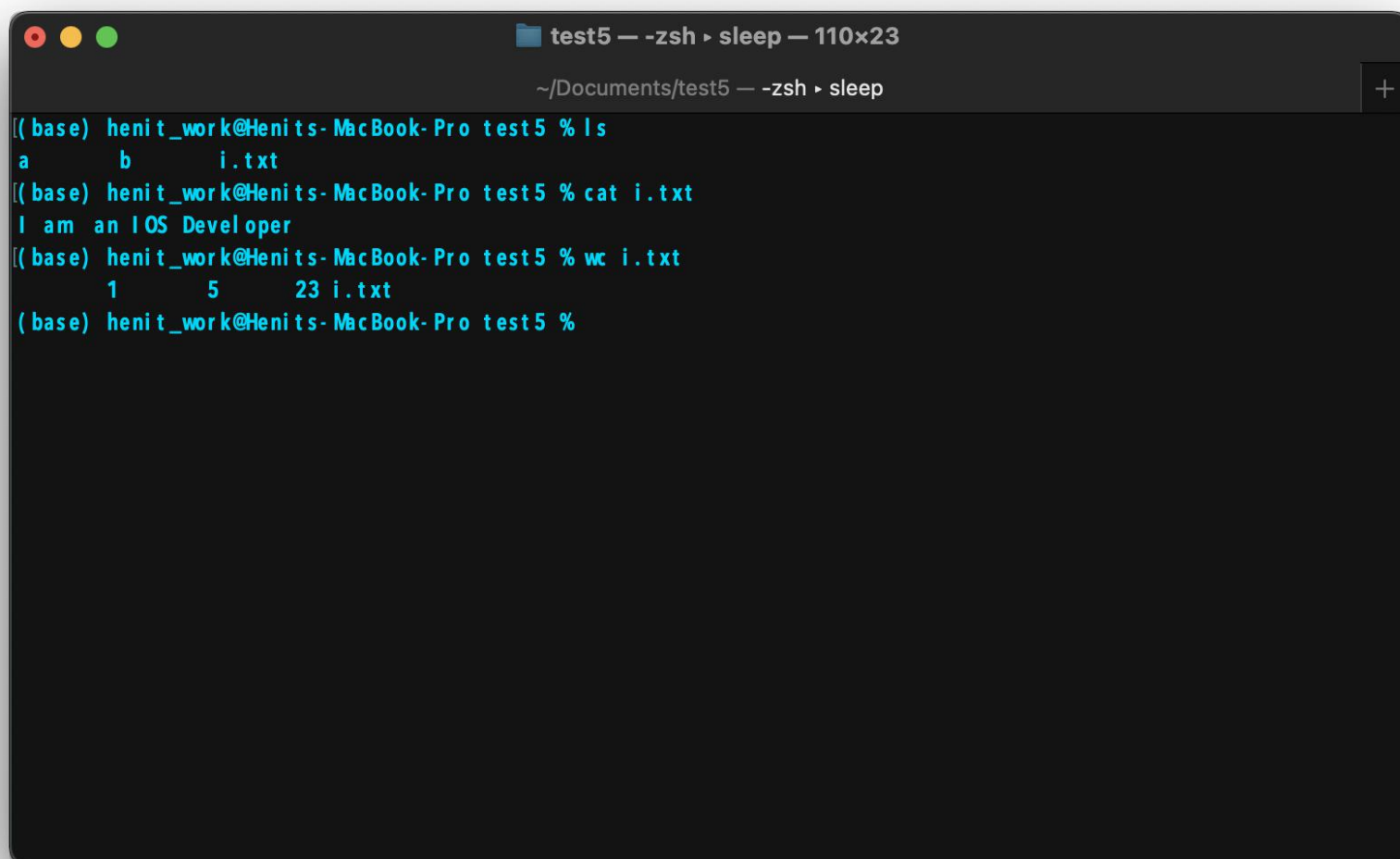
stat represents status, stat followed by file name will provide the current status of a particular file.

```
test5 — -zsh ▸ sleep — 110x23
~/Documents/test5 — -zsh ▸ sleep

(base) henit_work@Henits-MacBook-Pro test5 % ls
a      b
(base) henit_work@Henits-MacBook-Pro test5 % touch output.png
(base) henit_work@Henits-MacBook-Pro test5 % curl https://i.pcmag.com/imagery/reviews/03ai zyl UVApdyLAI ku1AvRV-
39.1605559903.fit_scale.size_760x427.png --output output.png
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left   Speed
100 13387    0 13387    0    0  32257      0 --:--:-- --:--:-- --:--:--  32257
(base) henit_work@Henits-MacBook-Pro test5 % open output.png
(base) henit_work@Henits-MacBook-Pro test5 % ls
a      b      output.png
(base) henit_work@Henits-MacBook-Pro test5 % stat output.png
16777223 31386407 -rw-r--r-- 1 henit_work staff 0 13387 "Aug 25 21:24:11 2021" "Aug 25 21:24:01 2021" "Aug 25
21:24:10 2021" "Aug 25 21:23:49 2021" 4096 32 0 output.png
(base) henit_work@Henits-MacBook-Pro test5 %
```

## WC :

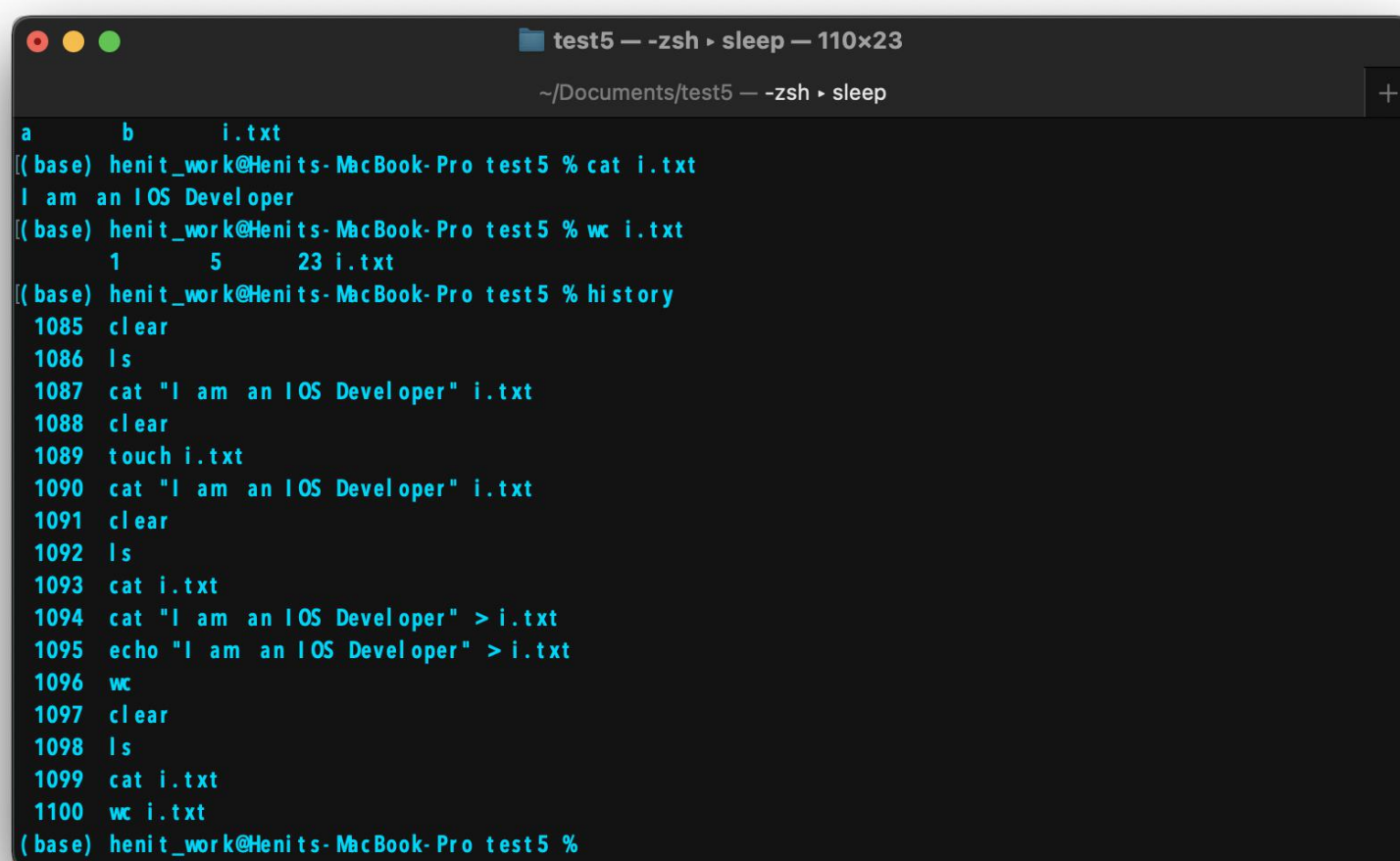
wc represents wordcount, execution of this command with a file name, will give the number of lines followed by number of words followed by number of letters.



```
test5 — -zsh ▸ sleep — 110×23
~/Documents/test5 — -zsh ▸ sleep
[(base) henit_work@Henits-MacBook-Pro test5 % ls
a      b      i.txt
[(base) henit_work@Henits-MacBook-Pro test5 % cat i.txt
I am an IOS Developer
[(base) henit_work@Henits-MacBook-Pro test5 % wc i.txt
      1      5     23 i.txt
(base) henit_work@Henits-MacBook-Pro test5 %
```

## history :

history lists out all the commands previously executed.

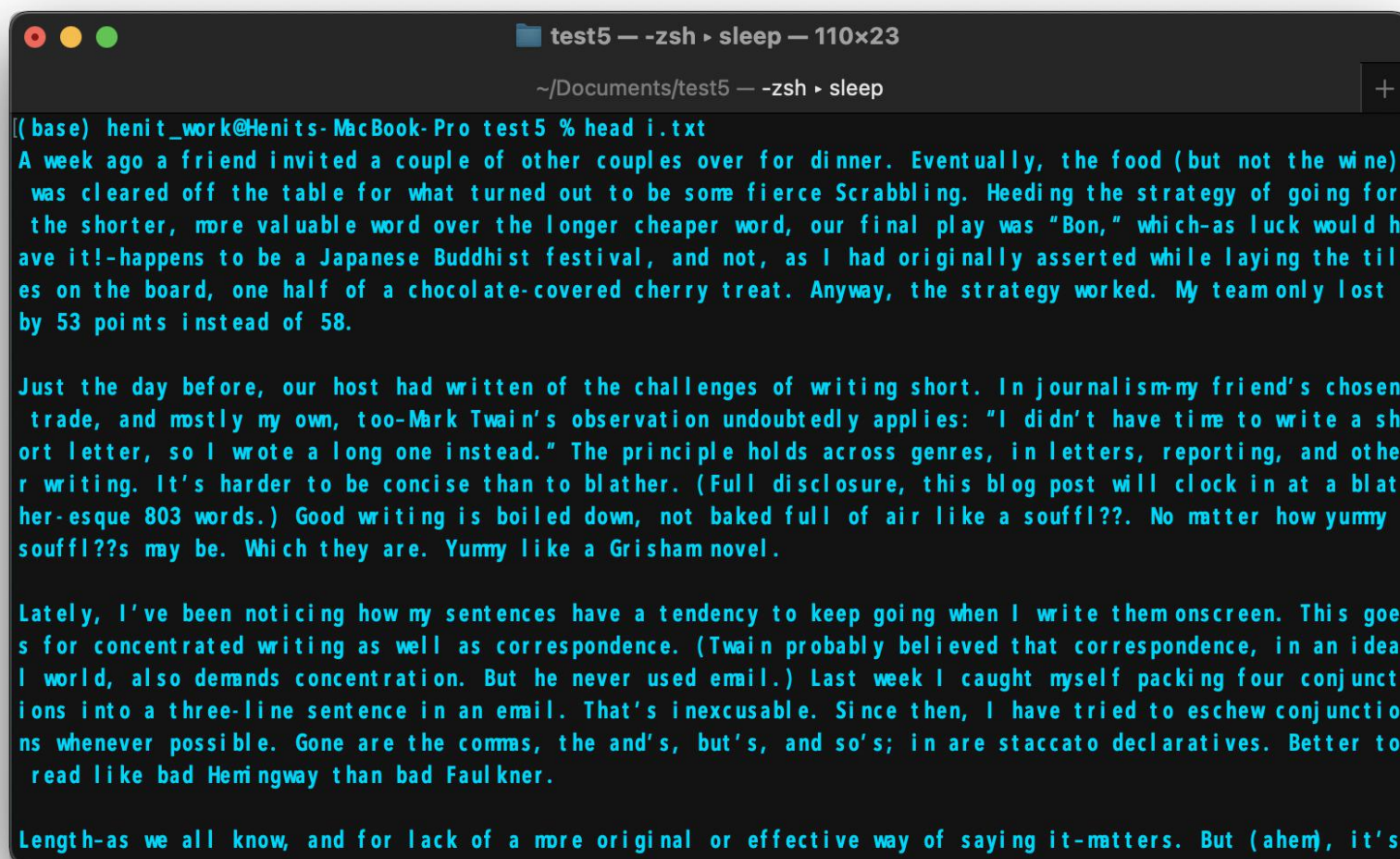


```
test5 — -zsh ▸ sleep — 110×23
~/Documents/test5 — -zsh ▸ sleep
a      b      i.txt
[(base) henit_work@Henits-MacBook-Pro test5 % cat i.txt
I am an IOS Developer
[(base) henit_work@Henits-MacBook-Pro test5 % wc i.txt
      1      5     23 i.txt
[(base) henit_work@Henits-MacBook-Pro test5 % history
1085 clear
1086 ls
1087 cat "I am an IOS Developer" i.txt
1088 clear
1089 touch i.txt
1090 cat "I am an IOS Developer" i.txt
1091 clear
1092 ls
1093 cat i.txt
1094 cat "I am an IOS Developer" > i.txt
1095 echo "I am an IOS Developer" > i.txt
1096 wc
1097 clear
1098 ls
1099 cat i.txt
1100 wc i.txt
(base) henit_work@Henits-MacBook-Pro test5 %
```

## head :

head prints the initail paragraph of a few lines of a particular files.





```
test5 — -zsh ▸ sleep — 110x23
~/Documents/test5 — -zsh ▸ sleep
[(base) henit_work@Henits-MacBook-Pro test5 % head i.txt
A week ago a friend invited a couple of other couples over for dinner. Eventually, the food (but not the wine)
  was cleared off the table for what turned out to be some fierce Scrabbling. Heeding the strategy of going for
  the shorter, more valuable word over the longer cheaper word, our final play was “Bon,” which-as luck would h
ave it!-happens to be a Japanese Buddhist festival, and not, as I had originally asserted while laying the til
es on the board, one half of a chocolate-covered cherry treat. Anyway, the strategy worked. My team only lost
by 53 points instead of 58.

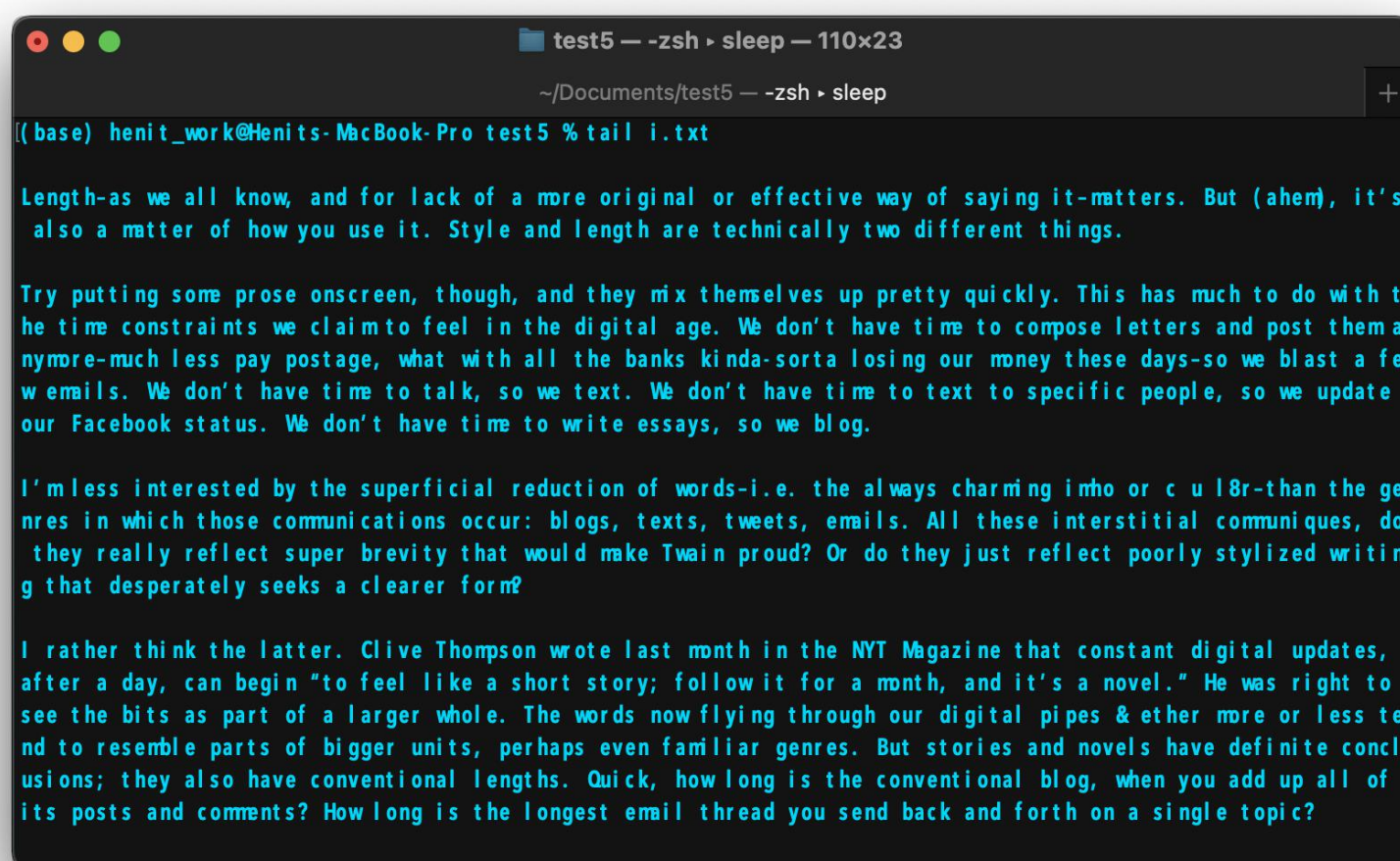
Just the day before, our host had written of the challenges of writing short. In journalism-my friend’s chosen
trade, and mostly my own, too-Mark Twain’s observation undoubtedly applies: “I didn’t have time to write a sh
ort letter, so I wrote a long one instead.” The principle holds across genres, in letters, reporting, and othe
r writing. It’s harder to be concise than to blather. (Full disclosure, this blog post will clock in at a blat
her-esque 803 words.) Good writing is boiled down, not baked full of air like a souffl??. No matter how yummy
souffl??s may be. Which they are. Yummy like a Grisham novel.

Lately, I’ve been noticing how my sentences have a tendency to keep going when I write them onscreen. This goe
s for concentrated writing as well as correspondence. (Twain probably believed that correspondence, in an idea
l world, also demands concentration. But he never used email.) Last week I caught myself packing four conjunct
ions into a three-line sentence in an email. That’s inexcusable. Since then, I have tried to eschew conjunctio
ns whenever possible. Gone are the commas, the and’s, but’s, and so’s; in are staccato declaratives. Better to
read like bad Hemingway than bad Faulkner.

Length-as we all know, and for lack of a more original or effective way of saying it-matters. But (ahem), it’s
```

**tail :**

tail prints couple of lines from the last part of file.



```
test5 — -zsh ▸ sleep — 110x23
~/Documents/test5 — -zsh ▸ sleep
[(base) henit_work@Henits-MacBook-Pro test5 % tail i.txt

Length-as we all know, and for lack of a more original or effective way of saying it-matters. But (ahem), it’s
also a matter of how you use it. Style and length are technically two different things.

Try putting some prose onscreen, though, and they mix themselves up pretty quickly. This has much to do with t
he time constraints we claim to feel in the digital age. We don’t have time to compose letters and post them a
nymore-much less pay postage, what with all the banks kinda-sorta losing our money these days-so we blast a fe
w emails. We don’t have time to talk, so we text. We don’t have time to text to specific people, so we update
our Facebook status. We don’t have time to write essays, so we blog.

I’m less interested by the superficial reduction of words-i.e. the always charming imho or c u l8r-than the ge
nres in which those communications occur: blogs, texts, tweets, emails. All these interstitial communiques, do
they really reflect super brevity that would make Twain proud? Or do they just reflect poorly stylized writin
g that desperately seeks a clearer form?

I rather think the latter. Clive Thompson wrote last month in the NYT Magazine that constant digital updates,
after a day, can begin “to feel like a short story; follow it for a month, and it’s a novel.” He was right to
see the bits as part of a larger whole. The words now flying through our digital pipes & ether more or less te
nd to resemble parts of bigger units, perhaps even familiar genres. But stories and novels have definite concl
usions; they also have conventional lengths. Quick, how long is the conventional blog, when you add up all of
its posts and comments? How long is the longest email thread you send back and forth on a single topic?
```

## Task 2 - Shell Scripts :

### Question 1 :

Consider two operands a and b which has value of 20 and 10 respectively write shell scripts to execute the relation operators such as -eq , -ne, -gt, -le.

```

echo "Hey! this is assignment 1"

a=20
b=10

if [ $a -eq $b ]
then
    echo "They both are equal"
else
    echo "Sorry they are not equal"
fi

if [ $a -le $b ]
then
    echo "a is less that b"
else
    echo " a is not less than b"
fi

if [ $a -gt $b ]
then
    echo "a is greater than b"
else
    echo "a is not greater than b"
fi

if [ $a -ne $b ]
then
    echo "a is not equal to b"
else
    echo " a is equal to b "
fi

```

**Output :**

```

(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1.sh
Hey! this is assignment 1
Sorry they are not equal
 a is not less than b
a is greater than b
a is not equal to b
(base) henit_work@Henits-MacBook-Pro OS Programming % |

```

## Ques 2

**Shell script to find the number is even or odd**

```

echo "Please enter the number"
read n

if [ `expr $n % 2` == 0 ]
then
    echo "$n is even"
else
    echo "$n is odd"
fi

```

```

(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-2.bash
Please enter the number
2
2 is even
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-2.bash
Please enter the number
5
5 is odd
(base) henit_work@Henits-MacBook-Pro OS Programming % |

```



## Ques 3

Write shell script to find the largest number among the five

```
i=1
max=0

echo "Enter Numbers"
while [ $i -le 5 ]
do
    read num
    if [ $i -eq 1 ] #set first number as max
    then
        max=$num
    else           #from number 2 update max if the num > max.
        if [ $num -gt $max ]
        then
            max=$num
        fi
    fi
    i=$((i + 1)) #increment i by 1
done

echo $max
```

```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
Enter Numbers
1
2
3
4
5
5
(base) henit_work@Henits-MacBook-Pro OS Programming % |
```

## Ques 4

Write a shell script to find the largest element in the array

```
echo "Enter Size(N)"
read N

i=1
max=0

echo "Enter Numbers"
while [ $i -le $N ]
do
    read num
    if [ $i -eq 1 ] #set first number as max
    then
        max=$num
    else           #from number 2 update max if the num > max.
        if [ $num -gt $max ]
        then
            max=$num
        fi
    fi
    i=$((i + 1)) #increment i by 1
done

echo $max
```



```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
Enter Size(N)
10
Enter Numbers
20
13
14
18
12
14
12
10
11
15
20
(base) henit_work@Henits-MacBook-Pro OS Programming %
```

## Ques 5

Write a shell script to display element A to Z in a loop

```
chars=( {a..z} )
n=3
for ((i=0; i<n; i++))
do
    echo "${chars[i]}"
done
```

```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
(base) henit_work@Henits-MacBook-Pro OS Programming %
```